



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/820,185	03/28/2001	David A. Zygmunt	15104.0001U2	3521
23859 7590 08/01/2007 NEEDLE & ROSENBERG, P.C. SUITE 1000 999 PEACHTREE STREET ATLANTA, GA 30309-3915			EXAMINER DAO, THUY CHAN	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 08/01/2007	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 09/820,185	<b>Applicant(s)</b> ZYGMONT ET AL.	
	<b>Examiner</b> Thuy Dao	<b>Art Unit</b> 2192	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 20 April 2007.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-40 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-40 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 28 March 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. In view of the Appeal Brief (hereinafter "Brief") filed on April 20, 2007, PROSECUTION IS HEREBY REOPENED. A new ground of rejection has been set forth below.

To avoid abandonment of the application, appellants must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below.

2. Claims 1-40 have been examined.

### **Response to Arguments**

3. The Applicants are thanked for a thorough reply. Applicants' arguments have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, an art made of record is being applied and new ground of rejection is made as set forth below.

### **Priority**

4. The instant application claims priority under Section 119 of US Provisional Application No. 60/192,431 (hereinafter "Provisional") filed on March 28, 2000. However, discrete disclosures have been found between the Provisional and Non-provisional applications. The Provisional consisted of:

page 1, Table of Contents;

page 2, Abstract;  
pp. 3-8: slides with brief titles, subsections, benefit description;  
pp. 9-13: table of contents (of an existing document or still in progress?)  
without any detailed description related to the claimed limitations;  
pp. 14-19: Figures without any detailed description related to the claimed  
limitations;  
pp. 20-21: listing of code, Metaprogram - class;  
pp. 22-24: listing of code, Metaprogram - component; and  
pp. 25-30: listing of code, Metaprogram – utility.

The examiner notes that the Provisional and Non-provisional disclosures (including specification and figures) are different, and the Provisional does not seem to fully support the claimed limitations, at least in independent claims 1, 18, 34, and 38, and which issues will be addressed hereto below.

The priority date for the instant application is being raised as whether it is in fact qualified to its entitled date from the Provisional application 60/192,431 (March 28, 2000) or the filing date of the instant application (March 28, 2001).

Accordingly, the priority date considered for the instant application is pending. The examiner treats the priority date as the filing date of the Provisional (March 28, 2000) and awaits response from the Applicants.

For example, priority – claims' mapping:

**Claim 1 and similar limitations in claim 18** called for particular limitations such as:

*... providing an object model expressed in an object modeling computer language, the object model representing a software system and comprising components realizing classes;*

*providing a set of one or more metaprograms reflecting a computer system architecture; and*

*a meta-machine binding the components to the metaprograms to generate the software system for a computer system having said architecture.*

Art Unit: 2192

The Provisional merely illustrated either in a block diagram (pp. 5-6) and/or listed in some sort of briefing format (pp. 3-4 and 7-8), subject matter or heading as table-of-content format (pp. 9-13).

The Provisional merely illustrated "Object Model" in one block of a diagram in page 5, which does not seem to support the claimed limitations "the object model representing a software system and comprising components realizing classes" (emphasis added).

The Provisional merely mentioned "Reuses application (model) across platforms (metaprograms)" in page 7, which does not seem to support the claimed limitations "a set of one or more metaprograms reflecting a computer system architecture" (emphasis added).

The Provisional merely listed "Binding" at page 9 (table-of-content format of an existing document or still in progress?), which does not seem to support the claimed limitations "a meta-machine binding the components to the metaprograms to generate the software system for a computer system having said architecture" (emphasis added).

**Claim 34 and similar limitations in claim 38** called for particular limitations such as:

... providing a meta-development environment (MDE) to a first party  
having access to a first computer system with a first architecture;

providing the MDE to a second party having access to a second computer  
system with a second architecture;

providing the first party's MDE with an object model expressed in an object  
modeling computer language, the object model representing a software system and  
comprising components realizing classes;

providing the second party's MDE with the object model;

providing the first party's MDE with a first set of one or more  
metaprograms reflecting the first architecture, wherein binding in the MDE of the  
components of the object model and the first set of metaprograms defines first software  
system code executable on a computer system having the first architecture but not  
executable on a computer system having the second architecture, the first software

Art Unit: 2192

system code effecting a user application when executed on the first computer system;  
and

*providing the second party's MDE with a second set of one or more metaprograms reflecting the second architecture, wherein binding in the MDE of the components of the object model and the second set of metaprograms defines second software system code executable on a computer system having the second architecture but not executable on a computer system having the first architecture, the second software system code effecting the same user application when executed on the second computer system as the first software system code effects when executed on the first computer system.*

After further consideration, the examiner could not find any written description about said particular claimed limitations. The Provisional does not seem to support the recited limitations "... providing a meta-development environment (MDE) to a first party ... to a second party ... providing the first party' MDE with ... providing the second party' MDE with ...,

*wherein binding in the MDE of the components of the object model and the first set of metaprograms defines first software system code executable on a computer system having the first architecture but not executable on a computer system having the second architecture, the first software system code effecting a user application when executed on the first computer system; and ...*

*wherein binding in the MDE of the components of the object model and the second set of metaprograms defines second software system code executable on a computer system having the second architecture but not executable on a computer system having the first architecture, the second software system code effecting the same user application when executed on the second computer system as the first software system code effects when executed on the first computer system (emphasis added).*

**Drawings**

Art Unit: 2192

5. The drawings are objected to because of minor informalities: hand-written text in drawings (e.g., FIG. 3-5F).

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

#### **Claim Objections**

6. Claims 6, 9, and 22 are objected to because of the following informalities: "Stereootype" should be - -Stereotype- -. Appropriate correction is required.

#### **Claim Rejections – 35 USC § 101**

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 18-33 are rejected because the claimed invention is directed to non-statutory subject matter: "A meta-development environment computer program product ...".

They amount to Functional Descriptive Material: "Data Structures" representing descriptive material per se or "Computer Programs" representing computer listings per se.

Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer. See, e.g., Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions. See MPEP 2106.

Under the principles of compact prosecution, claims 18-33 have been examined as the Examiner anticipates the claims will be amended to obviate these 35 USC § 101 issues. For example, - -A meta-development environment computer program product ... comprising a computer-readable medium [[carrying]] storing thereon: ...- -.



### Claim Rejections – 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

As noted above regarding priority date pending at issue, the examiner applies ground of rejection under 35 USC § 102(e).

10. Claim 1-40 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent No. 6,018,627 to Iyengar et al. (art made of record, hereinafter "Iyengar").

#### Claim 1:

Iyengar discloses *a method for developing a software system, comprising the steps of:*

*providing an object model expressed in an object modeling computer language, the object model representing a software system and comprising components realizing classes (e.g., FIG. 1, blocks 26, 27, 28, col.5: 17-41);*

*providing a set of one or more metaprograms reflecting a computer system architecture (e.g., col.9: 29-35; FIG. 10A-C, col.9: 49 – col.10: 50); and*

*a meta-machine binding the components to the metaprograms to generate the software system for a computer system having said architecture (e.g., FIG. 1, block 30 "Build and Wrap Components", block 31 "Assemble Applications and Components, block 32 "Deploy Applications", col.4: 12-26; FIG. 11-13: col.12: 1-44).*

**Claim 2:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML)* (e.g., col.2: 34-64).

**Claim 3:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the step of providing an object model is performed in part by a business analyst creating the object model; and the step of providing a set of one or more metaprograms is performed in part by a technologist coding the metaprograms* (e.g., col.3: 57 – col.4: 33).

**Claim 4:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the step of providing an object model comprises a user using a graphical user interface to list a project in a first window, the project representing the object model; and the step of providing one or more metaprograms comprises a user using a graphical user interface to list one or more metaprojects in a second window, each metaproject including a list of representations of the metaprograms* (e.g., FIG. 3, col.7: 29-49; FIG. 4, col.7: 50 – col.8: 20).

**Claim 5:**

The rejection of claim 4 is incorporated. Iyengar also discloses *the object modeling computer language includes an extension mechanism for the specification of user-defined properties and the assignment of these properties and their values to elements of the object model* (e.g., FIG. 2B, col.6: 1-38).

**Claim 6:**

The rejection of claim 5 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); and the user-defined extension mechanism is a Stereotype* (e.g., col.10: 51-58).

**Claim 7:**

The rejection of claim 5 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); and the user-defined extension mechanism is Tagged Values* (e.g., col.11: 33-47).

**Claim 8:**

The rejection of claim 5 is incorporated. Iyengar also discloses *the step of a meta-machine binding the components to the metaprograms comprises the steps of: searching the list of metaprojects for a metaproject having a name matching a name of an implementation target, wherein the implementation target is defined by the user-defined extension mechanism associated with the components; and storing an indication of an association between the metaproject having the matching name with metaprograms of which representations are listed in the metaproject having the matching name* (e.g., FIG. 5, col.8: 21-41).

**Claim 9:**

The rejection of claim 8 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); the user-defined extension mechanism is a Stereotype; and the Stereotype indicates the implementation target* (e.g., col.2: 34-64; col.10: 51-58).

**Claim 10:**

The rejection of claim 8 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); the user-defined extension mechanism is Tagged Values; and the Tagged Values indicate the implementation target* (e.g., col.11: 33-47).

**Claim 11:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the step of a user using a graphical user interface to invoke a metaprogram editor* (e.g., col.12: 18-29).

**Claim 12:**

The rejection of claim 1 is incorporated. Iyengar also discloses *each metaprogram in said set of metaprograms includes code and metacode, and the metacode generates a portion of the source code of the soft-ware system by outputting the code* (e.g., col.9: 53 – col.10: 3).

**Claim 13:**

The rejection of claim 12 is incorporated. Iyengar also discloses *the step of a user using a graphical user interface to invoke a metaprogram editor* (e.g., col.12: 18-29).

**Claim 14:**

The rejection of claim 13 is incorporated. Iyengar also discloses *the step of a user activating a toggling function of the metaprogram editor to toggle a window between highlighting the code and highlighting the metacode* (e.g., col.11: 16-25).

**Claim 15:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the set of metaprograms includes a model metaprogram that modifies the object model* (e.g., col.9: 49 – col.10: 15).

**Claim 16:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the set of metaprograms includes a component metaprogram invoked once for each component*

*and uses the classes realized by the component to produce a portion of the software system (e.g., col.12: 18-29; col.9: 53 – col.10: 3).*

**Claim 17:**

The rejection of claim 1 is incorporated. Iyengar also discloses *the set of metaprograms includes a class metaprogram that is invoked once for each class realized in each component and that produces a portion of the software system (e.g., col.9: 49 – col.10: 15).*

**Claim 18:**

Iyengar discloses *a meta-development environment computer program product for developing a software system, the computer program product comprising a computer-readable medium carrying thereon:*

*a meta-machine responsive to an object model and a set of one or more metaprograms input to the meta-machine under control of a user (e.g., FIG. 1, blocks 26-28, col.5: 17-41),*

*the object model expressed in an object modeling computer language (e.g., col.2: 34-64),*

*the object model representing a software system and comprising components realizing object classes (e.g., col.5: 17-41),*

*the set of metaprograms reflecting a computer system architecture (e.g., col.9: 29-35; FIG. 10A-C, col.9: 49 – col.10: 50),*

*the meta-machine under control of the user binding the components to the metaprograms to generate a resultant software system executable on a computer system having the architecture (e.g., FIG. 1, col.4: 12-26); and*

*a user interface for providing user control of metaprogram input, object model input, meta-machine operation, and resultant software system output (e.g., FIG. 11-13, col.12: 1-44).*

**Claim 19:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML)* (e.g., col.2: 34-64).

**Claim 20:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the user interface comprises a graphical user interface listing a project in a first window, the project representing the object model, and listing one or more metaprojects in a second window, each metaproject including a list of representations of the metaprograms* (e.g., col.7: 29-49; col.7: 50 – col.8: 20).

**Claim 21:**

The rejection of claim 20 is incorporated. Iyengar also discloses *the object modeling computer language includes an extension mechanism for the specification of user-defined properties and the assignment of these properties and their values to elements of the object model* (e.g., col.3: 57 – col.4: 33).

**Claim 22:**

The rejection of claim 21 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); and the user-defined extension mechanism is a Stereotype* (e.g., col.10: 51-58).

**Claim 23:**

The rejection of claim 21 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); and the user-defined extension mechanism is Tagged Values* (e.g., col.11: 33-47).

**Claim 24:**

The rejection of claim 21 is incorporated. Iyengar also discloses *the meta-machine binds the components to the metaprograms by searching the list of*

Art Unit: 2192

*metaprojects for a metaproject having a name matching a name of an implementation target and storing an indication of an association between the metaproject having the matching name with metaprograms of which representations are listed in the metaproject having the matching name, wherein the implementation target is defined by the user-defined extension mechanism associated with the components (e.g., col.8: 21-41).*

**Claim 25:**

The rejection of claim 24 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); the user-defined extension mechanism is a Stereotype; and the Stereotype indicates the implementation target (e.g., col.2: 34-64; col.10: 51-58).*

**Claim 26:**

The rejection of claim 24 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML); the user-defined extension mechanism is Tagged Values; and the Tagged Values indicate the implementation target (e.g., col.11: 33-47).*

**Claim 27:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the user interface includes a metaprogram editor (e.g., col.12: 18-29).*

**Claim 28:**

The rejection of claim 18 is incorporated. Iyengar also discloses *each metaprogram in said set of metaprograms includes code and metacode, and the metacode generates a portion of the source code of the software system by outputting the code (e.g., col.9: 53 – col.10: 3).*

**Claim 29:**

The rejection of claim 28 is incorporated. Iyengar also discloses *the user interface includes a metaprogram editor (e.g., col.9: 53 – col.10: 3).*

**Claim 30:**

The rejection of claim 29 is incorporated. Iyengar also discloses *the metaprogram editor includes a toggling means for toggling a window between highlighting the code and highlighting the metacode (e.g., col.11: 16-25).*

**Claim 31:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the set of metaprograms includes a model metaprogram that modifies the object model (e.g., col.9: 49 – col.10: 15).*

**Claim 32:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the set of metaprograms includes a component metaprogram invoked once for each component and uses the classes realized by the component to produce a portion of the software system (e.g., col.12: 18-29; col.9: 53 – col.10: 3).*

**Claim 33:**

The rejection of claim 18 is incorporated. Iyengar also discloses *the set of metaprograms includes a class metaprogram that is invoked once for each class realized in each component and that produces a portion of the software system (e.g., col.9: 49 – col.10: 15).*

**Claim 34:**

Iyengar discloses *a method for developing software systems for a plurality of computer system architectures, comprising the steps of:*



Art Unit: 2192

*providing a meta-development environment (MDE) to a first party having access to a first computer system with a first architecture (e.g., FIG. 10A, block 185, col. 9: 49-67);*

*providing the MDE to a second party having access to a second computer system with a second architecture (e.g., FIG. 10A, block 195 or 205 or 215, col.9: 49 – col.10: 21);*

*providing the first party's MDE with an object model expressed in an object modeling computer language, the object model representing a software system and comprising components realizing classes (e.g., FIG. 10B, col.10: 1-44);*

*providing the second party's MDE with the object model (e.g., col.5: 17-41);*

*providing the first party's MDE with a first set of one or more metaprograms reflecting the first architecture, wherein binding in the MDE of the components of the object model and the first set of metaprograms defines first software system code executable on a computer system having the first architecture but not executable on a computer system having the second architecture, the first software system code effecting a user application when executed on the first computer system (e.g., FIG. 4, block 124, col.7: 50-65); and*

*providing the second party's MDE with a second set of one or more metaprograms reflecting the second architecture, wherein binding in the MDE of the components of the object model and the second set of metaprograms defines second software system code executable on a computer system having the second architecture but not executable on a computer system having the first architecture, the second software system code effecting the same user application when executed on the second computer system as the first software system code effects when executed on the first computer system (e.g., FIG. 5, col.8: 21-41; FIG. 7, col.8: 57 – col.9: 21).*

**Claim 35:**

The rejection of claim 34 is incorporated. Iyengar also discloses *the MDE includes a metaprogram editor and a meta-machine; the step of providing the first*

Art Unit: 2192

*party's MDE with a first set of one or more metaprograms includes the step of the first party using the metaprogram editor of the first party's MDE to write metaprogram code; and the step of providing the second party's MDE with a second set of one or more metaprograms includes the step of the second party using the metaprogram editor of the second party's MDE to write metaprogram code (e.g., col.7: 29-49; col.3: 57 – col.4: 33).*

**Claim 36:**

The rejection of claim 35 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML) (e.g., col.2: 34-64).*

**Claim 37:**

The rejection of claim 35 is incorporated. Iyengar also discloses *the step of providing the first party's MDE with an object model comprises the first party purchasing the object model from a vendor and loading the object model into the first party's MDE; and the step of providing the second party's MDE with an object model comprises the second party purchasing the same object model from a vendor and loading the object model into the second party's MDE (e.g., col.12: 18-29; col.9: 53 – col.10: 3).*

**Claim 38:**

Iyengar discloses *a method for developing software systems for a plurality of computer system architectures, comprising the steps of:*

*providing a meta-development environment (MDE) to a first party having access to a first computer system having an architecture (e.g., FIG. 10A, block 185, col.9: 49-67);*

*providing the MDE to a second party having access to a second computer system having the architecture (e.g., FIG. 10A, block 195 or 205 or 215, col.9: 49 – col.10: 21);*

*the first party writing a first object model expressed in an object modeling computer language and loading the first object model into the first party's MDE (e.g., FIG. 10B, col.10: 1-44),*

*the object model representing a first software system and comprising components realizing classes (e.g., col.5: 17-41);*

*the second party writing a second object model expressed in an object modeling computer language and loading the second object model into the second party's MDE, the object model representing a second software system and comprising components realizing classes (e.g., FIG. 1, col.5: 1-67);*

*the first party purchasing from a vendor a set of one or more metaprograms reflecting the architecture and loading the set of metaprograms into the first party's MDE (e.g., FIG. 5, col.8: 21-41),*

*wherein binding in the MDE of the components of the first object model and the set of metaprograms defines first software system code executable on a computer system having the architecture but not executable on a computer system having another architecture (e.g., FIG. 1, blocks 31-32, col.4: 12-26),*

*the first software system code effecting a first user application when executed on the first computer system (e.g., FIG. 11-13, col.12: 1-44); and*

*the second party purchasing from a vendor a set of one or more metaprograms reflecting the architecture and loading the set of metaprograms into the second party's MDE, wherein binding in the MDE of the components of the second object model and the set of metaprograms defines second software system code executable on a computer system having the architecture but not executable on a computer system having another architecture (e.g., FIG. 4, block 124, col.7: 50-65, FIG. 10B, col.10: 1-44),*

*the second software system code effecting a second user application when executed on the second computer system different from that which the first software system code effects when executed on the first computer system (e.g., FIG. 10A, col.9: 49 – col.10: 21; col.9: 29-35; col.9: 49 – col.10: 21).*

**Claim 39:**

The rejection of claim 38 is incorporated. Iyengar also discloses *the object modeling computer language is the Unified Modeling Language (UML) (e.g., col.2: 34-64).*

**Claim 40:**

The rejection of claim 38 is incorporated. Iyengar also discloses *the step of the first party writing a first object model comprises a business analyst writing the first object model; and the step of the second party writing a second object model comprises a business analyst writing the second object model (e.g., col.6: 1-38; col.8: 21-41).*

11. Claims 1, 18, 34, and 38 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent No. 6,199,195 to Goodwin et al. (art made of record, hereinafter "Goodwin").

**Claim 1:**

Goodwin discloses *a method for developing a software system, comprising the steps of:*

*providing an object model expressed in an object modeling computer language, the object model representing a software system and comprising components realizing classes (e.g., FIG. 2, blocks 202-206, col.6: 10-22; col.4: 21-49);*

*providing a set of one or more metaprograms reflecting a computer system architecture (e.g., FIG. 2, block 208, col.6: 18-36; col.7: 51 – col.8: 41; col.9: 7-26); and*

*a meta-machine binding the components to the metaprograms to generate the software system for a computer system having said architecture (e.g., FIG. 2, block 210-212, col.6: 52-63; FIG. 5, col.13; 36 – col.14: 9).*

**Claim 18:**

lyengar discloses a *meta-development environment computer program product for developing a software system, the computer program product comprising a computer-readable medium carrying thereon:*

*a meta-machine responsive to an object model and a set of one or more metaprograms input to the meta-machine under control of a user (e.g., col.6: 52-63; FIG. 5, col.13: 36 – col.14: 9),*

*the object model expressed in an object modeling computer language (e.g., col.6: 10-22),*

*the object model representing a software system and comprising components realizing object classes (e.g., col.4: 21-49),*

*the set of metaprograms reflecting a computer system architecture (e.g., col.6: 18-36; col.9: 7-26),*

*the meta-machine under control of the user binding the components to the metaprograms to generate a resultant soft-ware system executable on a computer system having the architecture (e.g., col.7: 51 – col.8: 41; col.6: 52-63); and*

*a user interface for providing user control of metaprogram input, object model input, meta- machine operation, and resultant software system output (e.g., col.13: 36 – col.14: 9; col.6: 37-63).*

**Claim 34:**

lyengar discloses a *method for developing software systems for a plurality of computer system architectures, comprising the steps of:*

*providing a meta-development environment (MDE) to a first party having access to a first computer system with a first architecture (e.g., FIG. 5, blocks 508-510 and related text; col.7: 63 – col.8: 41);*

*providing the MDE to a second party having access to a second computer system with a second architecture (e.g., col.8: 42 – col.9: 27);*

*providing the first party's MDE with an object model expressed in an object modeling computer language, the object model representing a software system and comprising components realizing classes (e.g., col.11: 8-26);*

*providing the second party's MDE with the object model (e.g., col.13: 20-38; col.14: 10-39);*

*providing the first party's MDE with a first set of one or more metaprograms reflecting the first architecture, wherein binding in the MDE of the components of the object model and the first set of metaprograms defines first software system code executable on a computer system having the first architecture but not executable on a computer system having the second architecture, the first software system code effecting a user application when executed on the first computer system (e.g., FIG. 2 and related text; col.14: 51 – col. 15: 34); and*

*providing the second party's MDE with a second set of one or more metaprograms reflecting the second architecture, wherein binding in the MDE of the components of the object model and the second set of metaprograms defines second software system code executable on a computer system having the second architecture but not executable on a computer system having the first architecture, the second software system code effecting the same user application when executed on the second computer system as the first software system code effects when executed on the first computer system (e.g., FIG. 5 and related text; col.15: 14-44; col.6: 37-63).*

**Claim 38:**

lyengar discloses a *method for developing software systems for a plurality of computer system architectures, comprising the steps of:*

*providing a meta-development environment (MDE) to a first party having access to a first computer system having an architecture (e.g., col.7: 63 – col.8: 41);*

*providing the MDE to a second party having access to a second computer system having the architecture (e.g., FIG. 2 and related text; col.11: 8-26);*

*the first party writing a first object model expressed in an object modeling computer language and loading the first object model into the first party's MDE (e.g., col.9: 32 – col.10: 18),*

*the object model representing a first software system and comprising components realizing classes (e.g., col.6: 10-22; col.4: 21-49);*

Art Unit: 2192

*the second party writing a second object model expressed in an object modeling computer language and loading the second object model into the second party's MDE (e.g., col.6: 37-63; col.13: 45 – col.14: 8),*

*the object model representing a second software system and comprising components realizing classes (e.g., FIG. 2 and related text; col.15: 14-44);*

*the first party purchasing from a vendor a set of one or more metaprograms reflecting the architecture and loading the set of metaprograms into the first party's MDE (e.g., FIG. 5 and related text; col.6: 18-36),*

*wherein binding in the MDE of the components of the first object model and the set of metaprograms defines first software system code executable on a computer system having the architecture but not executable on a computer system having another architecture (e.g., col.7: 51-41; col.9: 7-26),*

*the first software system code effecting a first user application when executed on the first computer system (e.g., col.6: 52-63; col.13: 36 – col.14: 9); and*

*the second party purchasing from a vendor a set of one or more metaprograms reflecting the architecture and loading the set of metaprograms into the second party's MDE, wherein binding in the MDE of the components of the second object model and the set of metaprograms defines second software system code executable on a computer system having the architecture but not executable on a computer system having another architecture (e.g., col. 14: 10-39; col.14: 51 – col.15: 34),*

*the second software system code effecting a second user application when executed on the second computer system different from that which the first software system code effects when executed on the first computer system (e.g., FIG. 2, 5 and related text; col.9: 32 – col.10: 18; col.11: 8-26).*

### **Conclusion**

12. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

Art Unit: 2192

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao



TUAN DAM  
SUPERVISORY PATENT EXAMINER